

## **Remarks**

These Remarks are in reply to the Office Action mailed February 7, 2008.

### **I. Summary of Rejections**

The Office Action rejected claims 11-21.

Claims 11-20 were rejected under 35 U.S.C. 102(b) as being anticipated by Bogle et al. (U.S. Pat. No. 6,353,923 B1).

Claim 21 was rejected under 35 U.S.C. 103(a) as being unpatentable over Bogle et al. (U.S. Pat. No. 6,353,923 B1).

### **II. Related Application No. 10/784,600 Was Allowed**

Applicants bring to the attention of the Examiner that the related U.S. Application No. 10/784,600 was allowed. The Examiner's statement of reasons for allowance in application 10/784,600 stated, "Bogle and other cited prior art, taken alone or in combination failed to disclose limitations as cited..."

### **III. The Prior Art Reference Does Not Anticipate or Suggest Claims 11-21**

#### **35 U.S.C. §102 Rejection to Independent Claim 11**

Independent Claim 11 states:

A system for debugging in more than one programming language, comprising:

a multi-language debugger with the capability to debug a source code file which contains multiple nested languages, wherein the multi-language debugger interprets multiple languages that are nested in a single source file, and wherein the multiple nested languages can include both

compiled and interpreted languages;

a script debug controller, wherein the multi-language debugger uses a standardized interface for a script engine, wherein all communications with the script engine will be through calls to the script debug controller;

a debuggable frame object, wherein the script engine uses a debuggable frame object to retrieve script context for a supported language, wherein each of the multiple nested languages is displayed in a debuggable frame object, and wherein each of the multiple nested languages can be edited in the debuggable frame object;

an interface to a messaging environment, wherein the interface is implemented by a runtime messaging environment that controls a running state of the script engine; and

a debug commands interface.

Claim 11 includes “a multi-language debugger with the capability to debug a source code file which contains multiple nested languages, wherein the multi-language debugger interprets multiple languages that are nested in a single source file, and wherein the multiple nested languages can include both compiled and interpreted languages.” This feature is not shown or made obvious by the cited reference.

The Office Action presented on pages 2-3 the erroneous argument that all that is necessary to anticipate Claim 11 is a “multi-language debugger,” not “a multi-language debugger with the capability to debug a source code file which contains multiple nested languages, wherein the multi-language debugger interprets multiple languages that are nested in a single source file, and wherein the multiple nested languages can include both compiled and interpreted languages.” The Office Action incorrectly argued that all the language after “a multi-language debugger” did not result in a

patentable change.

The Office Action further incorrectly relied upon *Gardner v. TEC Systems, Inc.*, 725 F.2d 1338, 220 USPQ 777 (Fed. Cir. 1984). In *Gardner*, the only difference between the prior art air bar and the claimed air bar was the size of the claimed air bar, and the air bar did not perform or operate any differently from the prior art air bar. *Gardner* does not support the Office Action's position, as there clearly is a difference between a debugger that is capable of debugging a single source code file that contains multiple nested compiled and interpreted languages, and a debugger that cannot debug a single source code file that contains multiple nested compiled and interpreted languages.

Bogle's invention is focused on debugging an application that includes multiple program components from many different programming language sources, i.e. many different source code files, not a single source code file as in claim 11 (see Bogle, col. 2, line 1: "many different programming language sources"). Bogle's invention simply does not do and does not claim to do what Applicants' claim 11 can do.

The Office Action cited Bogle's FIG. 4 and the statement in col. 4, lines 10 – 19, for disclosing this portion of claim 11. FIG. 4 shows multiple host processes and the debug managers, FIG. 4 does not indicate the capability to debug multiple nested languages in a single source code file. Bogle's col. 4, lines 10 – 19, describes a method for debugging a virtual application that includes multiple compiled and interpreted programming language statements. However, col. 4 says nothing regarding an ability to debug multiple nested languages in a single source code file. There is a distinction between debugging a virtual application that contains program code in several source code files versus debugging a single source code file that contains multiple nested languages.

Furthermore, Bogle's col. 11, lines 25-37 describes Bogle's Machine Debug Manager in greater detail.

“Virtual applications are collections of related documents and code in a single debuggable entity such that separate application components in a continuous line of code can share a common process and/or thread. A virtual application is the aggregate of multiple applications in multiple programming languages. One key role of the machine debug manager is to act as a program language registry that provides a mapping between a given application in the virtual application aggregate and the active debugger IDE 410 that is controlling the virtual application during the debug process. The MDM 411 eliminates the traditional debugging model where the debugger for a given programming language only has knowledge of a specific source and object code mapping.”

This portion of Bogle clearly indicates that Bogle saw his application as not focused on a specific source, i.e. not focused on a specific source code file, and instead Bogle’s application focused on the virtual application as a whole, as collections of related documents and source code files were treated as a single application entity in the aggregate.

As Applicants stated on page 4 of the Detailed Description, “[A] developer can debug source code that has several nested languages within a single source file. Mixing several languages in a single source file is becoming an increasingly valuable and popular capability.”

Applicants respectfully submit that the embodiment as defined in independent claim 11 is neither anticipated by nor obvious in view of Bogle.

### 35 U.S.C. §102 Rejection to Independent Claim 15

Claim 15 requires “a proxy, wherein the proxy is used between the executing code being debugged and the debugger.” The Office Action cited Bogle’s marshalling proxy as disclosing the features of claim 15. Bogle does not teach or suggest that the marshalling proxy is used between the executing code being debugged and the debugger.

#### Rejections to Claims 12-21

For at least the reasons discussed above, dependent claims 12-21 are also patentable. Dependent claims 12-21 add their own limitations which render them patentable in their own right.

#### IV. Conclusion

In light of the above, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and a Notice of Allowance is requested. The Examiner is respectfully requested to telephone the undersigned before an advisory action is issued in order to avoid any unnecessary filing of an appeal.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: April 4, 2008

By: /Thomas K. Plunkett/  
Thomas K. Plunkett  
Reg. No. 57,253

Customer No. 23910  
FLIESLER MEYER LLP  
650 California Street, 14<sup>th</sup> Floor  
San Francisco, California 94108  
Telephone: (415) 362-3800